

PRIORITY DOCUMENT
CERTIFIED COPY OF 日本国特許庁
PRIORITY DOCUMENT JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年 2月19日

出願番号

Application Number:

特願2001-041120

出願人

Applicant(s):

株式会社日立製作所

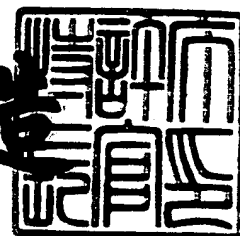
日立ソフトウェアエンジニアリング株式会社

BEST AVAILABLE COPY

2001年 9月28日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3089310

【書類名】 特許願

【整理番号】 K00017581A

【提出日】 平成13年 2月19日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/00

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 関 由美子

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 亀田 正美

【発明者】

【住所又は居所】 神奈川県横浜市中区尾上町 6 丁目 8 1 番地 日立ソフトウェアエンジニアリング株式会社内

【氏名】 藤井 剛

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【特許出願人】

【識別番号】 000233055

【氏名又は名称】 日立ソフトウェアエンジニアリング株式会社

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 システム管理支援方法及び装置

【特許請求の範囲】

【請求項 1】

サービスを実行する複数の計算機から構成されたシステムにおけるサービスの運用を決定する管理プログラムであって、

サービスの運用を定めた複数の運用ルールと、それぞれの上記運用ルール毎に当該運用ルールを実行するための複数の条件とを有し、

サービスの実行により生成されたログ情報をそれぞれの前記計算機から収集し、収集した前記ログ情報が条件を満たした場合に、当該条件に対応付けられた運用ルールに基づいて前記計算機を制御する命令を出力する管理プログラム。

【請求項 2】

請求項 1 記載の管理プログラムにおいて、

前記運用ルールは、サービスの要求を前記計算機に振分けるルール、前記条件は計算機の稼働率であって、

それぞれの計算機から収集した前記ログ情報から前記計算機の稼働率を求め、求めた当該稼働率が前記条件を満足する運用ルールを実行するための命令を出力する管理プログラム。

【請求項 3】

請求項 2 記載の管理プログラムにおいて、

求めた前記稼働率が前記条件を満足する運用ルールを表示し、当該運用ルールを実行するか否かを問合せ、当該運用ルールを実行するとの入力があった場合、当該運用ルールを実行するための命令を出力する管理プログラム。

【請求項 4】

複数のプログラムを有する計算機で実行されるプログラムの運用を決定する管理プログラムであって、

プログラムの運用を定めた複数のプログラム運用ルールと、それぞれの上記プログラム運用ルールに対応付けられた当該プログラム運用ルールを実行するための複数の条件とを有し、

実行されたプログラムのログ情報を収集し、収集した前記ログ情報が条件を満たした場合に、当該条件と対応付けられたプログラム運用ルールを前記計算機で実行されるようにする管理プログラム。

【請求項 5】

請求項 4 記載の管理プログラムにおいて、

収集した前記ログ情報から、計算機で実行されたプログラムの実行状態を表示し、収集した前記ログ情報が満たした条件と対応するプログラム運用ルールを実行するか否かの問合せを表示し、当該問合せに対して当該プログラム運用ルールを実行するとの入力があった場合に、当該プログラム運用ルールを前記計算機で実行されるようにする管理プログラム。

【請求項 6】

請求項 5 記載の運用管理プログラムにおいて、

表示される前記プログラムの実行状態は、収集した前記ログ情報に含まれる開始時刻、終了時刻をプログラム毎に表示する管理プログラム。

【請求項 7】

請求項 4 記載の運用管理プログラムにおいて、

前記プログラムの運用ルールは、実行されたプログラムと次に実行されるプログラムとを定義した管理プログラム。

【請求項 8】

サービスを提供する複数の計算機が接続されたシステムにおける計算機の管理方法であって、

少なくとも一つの計算機は、サービスの運用を定めた複数の運用ルールと、それぞれの上記運用ルール毎に当該運用ルールを実行するための複数の条件とを有し、

サービスの実行により生成されたログ情報をそれぞれの前記計算機から収集し、収集した前記ログ情報が条件を満たした場合に、当該条件に対応付けられた運用ルールを実行するための命令を出力する計算機の管理方法。

【請求項 9】

複数のプログラムを有する計算機で実行されるプログラムの運用を制御する制

御方法であって、

プログラムの運用を定めた複数のプログラム運用ルールと、それぞれの上記プログラム運用ルールに対応付けられた当該プログラム運用ルールを実行するための複数の条件とを有し、

実行されたプログラムのログ情報を収集し、収集した前記ログ情報が条件を満たした場合に、当該条件と対応付けられたプログラム運用ルールを前記計算機で実行する制御方法。

【請求項 1 0】

サービスを提供する複数の計算機が接続されたシステムであって、

少なくとも一つの計算機は、サービスの運用を定めた複数の運用ルールと、それぞれの上記運用ルール毎に当該運用ルールを実行するための複数の条件とを有し、

サービスの実行により生成されたログ情報をそれぞれの前記計算機から収集し、収集した前記ログ情報が条件を満たした場合に、当該条件に対応付けられた運用ルールを実行するシステム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、複数の計算機で実行されるサービス又は計算機で実行される複数のプログラムの運用管理に関するものである。

【0 0 0 2】

【従来の技術】

従来、サービスを提供するコンピュータがネットワークに接続されたサービス提供システムとして、例えば、特開平 1 1 - 1 0 2 3 3 6 号公報に開示された技術がある。ここでは、クライアントプログラム実行時の通信接続ログ情報を用いて、プログラム実行時に最大回数の接続を行うサーバコンピュータを決定し、そのサーバコンピュータにプログラムを移動して実行させる技術が開示されている。

【0 0 0 3】

【発明が解決しようとする課題】

上記従来技術には、上記開示された内容によりサーバコンピュータにかかる負荷を軽減し、ネットワーク接続回数の削減、ネットワークトラフィックの削減を実現することが開示されている。

一方、要求に対するサービスをより早く提供したり、要求に対する回答の信頼性を高くしたりするようにシステム全体を最適に実行することが必要である。このために、サービスの要求を振分けたり、コンピュータで実行されるサービスの実行順序（スケジューリング）を決定することが必要になる。この点については上記従来技術には開示されていない。

【0004】

また、このようにサービスの要求を振分けたり、コンピュータで実行されるサービスのスケジューリングを行う場合には、プログラムの変更も少なくなる。本発明は、既存のアプリケーション全体を作り直すことなく、日々変化する分散サービスに適応して最適実行を行うことのできる分散システム、管理方法及びプログラムを提供することにある。

【0005】

【課題を解決するための手段】

上記目的を達成するための一つの手段として、サービスの運用を定めた複数の運用ルールと、それぞれの運用ルール毎に運用ルールを実行するための複数の条件とを設定し、サービスの実行により生成されたログ情報をそれぞれの計算機から収集し、収集したログ情報が条件を満たした場合に、この条件に対応付けられた運用ルールが実行されるようにする。

【0006】

このようにログ情報に基づいて、サービスの運用が決定されるので、日々変化するサービスの要求に対応することが可能となり、最適にシステムを運用することができる。

【0007】

【発明の実施の形態】

以下、本発明の一実施形態について図面を用いて説明する。

【0008】

図1は、本実施例におけるシステム構成図である。本システムは、複数のデータベース101～103、複数のジョブ実行サーバ104、105、複数のアプリケーションサーバ106、107、分散システム管理サーバ108とがネットワークを介して接続されている。アプリケーションサーバ106、107は、複数のアプリケーションプログラムを有しており、必要に応じてデータベースへデータの格納、更新またはデータの読み出しを行う命令をジョブ実行サーバ104又は105へ送る。ジョブ実行サーバ104又は105は、アプリケーションサーバ106、107から送られてきた命令に従って、データベース101～103へアクセスし、データの格納、更新、読み出しを行う。分散システム管理サーバ108は、ジョブ実行サーバ104、105、アプリケーションサーバ106、107からログ情報を取り込み、ジョブ実行サーバやアプリケーションサーバの運用管理を行う。具体的には、ジョブ実行サーバ内あるいはアプリケーションサーバ内のジョブやアプリケーションの実行順序を制御したり、アプリケーションサーバからアクセスするジョブ実行サーバを制御したりするための運用ルールを生成する。

【0009】

ジョブ実行サーバは、CPUなどの処理部、ネットワークへの接続処理を行う通信処理部、光ディスクや磁気ディスクなどの記憶部、CRTや液晶ディスプレイなどの表示部、キーボードやマウスなどの入力部から構成されている。尚、アプリケーションサーバ、分散管理サーバも同様の構成である。

【0010】

図2は、分散システム管理サーバ108の記憶部に記憶されている主なデータ又はプログラムなどを示したものである。

【0011】

ログ情報201、ログ書式情報202は、ジョブ実行サーバ104、105、アプリケーションサーバ106、107のログ情報、ログ書式情報が格納されている。サーバアドレス203は、ジョブ実行サーバ、アプリケーションサーバのアドレスが格納されている。アプリケーション実行ルール204は、ジョブ実行

サーバ又はアプリケーションサーバの運用ルールあるいは、ジョブ実行サーバ又はアプリケーションサーバのジョブプログラムまたはアプリケーションプログラムの運用ルールが格納されている。分析データ 2 0 5 は、ログ情報 2 0 1 を解析処理した結果であり、ジョブ実行サーバ、アプリケーションサーバなどの運用実績やプログラムの運用実績などが格納されている。

【 0 0 1 2 】

ログ検出処理 2 0 6 は、ジョブ実行サーバ、アプリケーションサーバからログ情報、ログ書式情報を取得するためのプログラムである。ログデータ抽出処理 2 0 7 は、ログ書式情報に従ってログ情報を抽出するためのプログラムである。加工処理 2 0 8 は、抽出されたログデータを編集、加工するためのプログラムである。分析処理 2 0 9 は、編集、加工された結果を分析したり、分析結果を表示部に表示したりするためのプログラムである。実行ルール生成処理 2 1 0 は、提示された分析結果を基に最適実行のための最適化ルール（運用ルール）を生成するためのプログラムである。最適化実行処理 2 1 1 は、生成された最適化ルールとログ情報に従ってジョブ実行サーバ又はアプリケーションサーバを制御するための制御コマンド選択し、送るためのプログラムである。このように、分散システム管理サーバ 1 0 8 は、ジョブ実行サーバ 1 0 4、1 0 5、アプリケーションサーバ 1 0 6、1 0 7 からログ情報を収集し、ジョブプログラム又はアプリケーションプログラムを制御するための最適化ルールを生成する。更に生成された最適化ルールとログ情報によって制御コマンドを送ることで、直接ジョブ実行サーバ 1 0 4、1 0 5、アプリケーションサーバ 1 0 6、1 0 7 の運用を制御する。このようにすることで、ジョブ実行サーバ又はアプリケーションサーバでは、最適化ルールを実行するための特別なプログラムが不要となる。

【 0 0 1 3 】

尚、最適化ルールを実行するための特別なプログラムをジョブ実行サーバ、アプリケーションサーバに格納し、分散システム管理サーバは、最適化ルールを送るようにしてもよい。このようにすれば、ジョブ実行サーバ、アプリケーションサーバ内で最適化ルールに従った運用を行うことができるので、処理を高速にすることができる。

【0014】

例えば、ジョブ実行サーバ、アプリケーションサーバで高速な処理が要求されている場合には、最適化ルールを送るようにし、高速な処理が要求されていない場合には、分散システム管理サーバから制御コマンドを送るようにすれば、経済的な分散システムの環境を提供することができる。

アプリケーション212、213は、他の処理を実行するためのアプリケーションプログラムである。尚、ログ検出処理206、ログデータ抽出処理207、加工処理208、分析処理209、実行ルール生成処理210、最適化実行処理211は、既に説明したように別々のプログラムであってもよいし、これらをまとめて一つのプログラムとしてもよい。尚、それぞれの処理については後述する。

図3はアプリケーションサーバの記憶部に格納されている主なデータ又はプログラムを示したものである。ログ情報301、ログ書式情報302は、アプリケーションサーバのログ情報と、そのログ情報の書式を定義したログ書式情報が格納されている。アプリケーション実行ルール303は、分散システム管理サーバ108から送られてきた最適化ルールが格納されている。尚、先ほど説明したように、分散システム管理サーバから制御コマンドが直接送られてくるようにしている場合には、最適化ルールがなくてもよい。アプリケーション実行処理305は、アプリケーションサーバに接続可能なクライアント端末から送られてきた要求に基づいてアプリケーションの実行を行うためのプログラムである。具体的には、クライアントのクライアント端末から送られてくる要求と実行するアプリケーションとを予め対応付けて管理し、要求に応じて対応するアプリケーションを実行する。データベース管理処理308は、データベースへのデータの格納、更新、データの読み出しを行う際に、ジョブ実行サーバへ、データベースに対する処理のコマンドを発生するための処理を行うためのプログラムである。尚、データベース管理処理308は、アプリケーションプログラムからの要求によって処理を行うこともできる。ログ情報取得処理306は、アプリケーションサーバで実行されたコマンドなどをログ情報としてログ情報301に格納したり、あるいは、分散システム管理サーバ108からの要求に基づいてログ情報やログ書式情報を送る処理を行うプログラムである。通信処理309は、他のアプリケーション

サーバ、ジョブ実行サーバ、分散システム管理サーバ等と通信を行うためのプログラムである。アプリケーション 3 0 7、3 1 0 はクライアントからの要求に対して実行するアプリケーションプログラムである。また、そのほかのデータについては 3 0 4 に格納されている。

【 0 0 1 5 】

アプリケーションサーバは、予め定められたログ書式情報に従って、処理中のログをログ情報として出力する。

【 0 0 1 6 】

図 4 はジョブ実行サーバの記憶部に格納されている主なデータ又はプログラムを示したものである。

【 0 0 1 7 】

ログ情報 4 0 1、ログ書式情報 4 0 2 は、ジョブ実行サーバのログ情報と、そのログ情報の書式を定義したログ書式情報が格納されている。ジョブ実行ルール 4 0 3 は、分散システム管理サーバ 1 0 8 から送られてきた最適化ルールが格納されている。尚、先ほど説明したように、分散システム管理サーバから制御コマンドが直接送られてくるようにしている場合には、最適化ルールがなくてもよい。ジョブ実行処理 4 0 5 は、アプリケーションサーバから送られてきた要求に基づいてジョブの実行を行うためのプログラムである。具体的には、アプリケーションサーバから送られてくる要求と実行するジョブとを予め対応付けて管理し、要求に応じて対応するジョブを実行する。データベース管理処理 4 0 8 は、データベースへのデータの格納、更新、データの読み出しを行うためのプログラムである。ログ情報取得処理 4 0 6 は、ジョブ実行サーバで実行されたコマンドなどをログ情報としてログ情報 4 0 1 に格納したり、あるいは、分散システム管理サーバ 1 0 8 からの要求に基づいてログ情報やログ書式情報を送る処理を行うプログラムである。通信処理 3 1 0 は、他のアプリケーションサーバ、ジョブ実行サーバ、分散システム管理サーバ等と通信を行うためのプログラムである。ジョブ 4 0 7、4 1 0 はアプリケーションサーバからの要求に対して実行するジョブプログラムである。また、そのほかのデータについては 4 0 4 に格納されている。ジョブ実行サーバは、予め定められたログ出力書式設定情報に従って、処理中の

ログをログ情報として出力する。

【0018】

次に、分散システム管理サーバ108の処理について説明する。

【0019】

まず、サーバからログ情報を取得するログ検出処理206について説明する。ログ検出処理は、ログ情報を取得するためのリストの生成と、リストに従ってログ情報を取得する処理がある。

【0020】

図5は、リスト生成処理を示したものである。このリストは、ログ情報を収集するために利用するものである。

【0021】

処理511でログ情報の取得の対象となるプログラム名とそのプログラムが実行されるサーバの指定、ログ情報を収集する時間であるスケジュールを受け付ける。処理512でサーバの指定があるか否かを判断し、サーバの指定がある場合には処理513でサーバへ接続し、ログ情報の位置をサーバから取得してログリスト（図23参照）を生成する。処理512でサーバの指定がない場合には、サーバアドレスに従ってサーバに接続する。処理515で接続したサーバからプログラム名称を読み出し、指定されたプログラム名と一致した場合、処理517でこのプログラム名のログ情報の位置をサーバから取得し、ログリスト（図23参照）を生成する。次に、サーバアドレスでまだ接続されていないサーバがあると処理518で判断された場合、処理514へ戻る。また、処理516で一致するプログラム名がないと判断された場合にも処理514へ戻る。このように処理514～518はサーバアドレスに登録されているサーバに対して順次実行される。

尚、処理513、517のログリストの生成において、スケジュールが指定されている場合には、ログ情報の位置とともにログリストに書き込まれる。

【0022】

ここで、サーバであるそれぞれのジョブ実行サーバ、アプリケーションサーバは、記憶部に格納されているプログラム名称と、それぞれのプログラムのログ情

報、ログ書式情報の格納位置（アドレス或いはファイル名）を対応付けて記憶し、分散システム管理サーバからプログラム名称の要求を受け付けた場合には、記憶部に格納されているプログラム名称の一覧を出力する。また、分散システム管理サーバ 1 0 8 からログ情報、ログ書式情報の格納位置の要求を受け付けた場合には、ログ情報、ログ書式情報の格納位置を出力する。

【 0 0 2 3 】

図 2 3 は、ログリストの一例を示したものである。この図に示されるように装置（サーバ）のアドレスとログ情報名、対応するアプリケーション名、取り込み時間によってリストが生成される。ここで取り込み時間は、スケジュールとして受け付けた時間である。スケジュールを受け付けていない場合には、デフォルトが設定される。分散システム管理サーバは、このリストに基づいて各サーバからログ情報を取得する。ここでは、ログ情報の格納位置がファイル名である場合を示したが、ログ情報が格納されているアドレスであってもよい。つまり、分散システム管理サーバ 1 0 8 が、後にログ情報を収集できるようになっていればよい。

【 0 0 2 4 】

図 2 0 はログ情報取得処理を示したものである。処理 2 0 1 1 でログリストの取り込み時間を収集する。取り込み時間と分散システム管理サーバ内のタイマにより、取り込み時間と一致すると判断された場合（2 0 1 2）、ログリストによってサーバへアクセスし、ログ情報を収集する（2 0 1 3）。取り込み時間と一致しない場合には、再度ログリストを取り込む。尚、取り込み時間がデフォルトの場合は、分散システム管理サーバ 1 0 8 に設定された時間間隔でログ情報が取得される。また、デフォルトの場合にはログ情報の取得を行わないように設定することもできる。

【 0 0 2 5 】

このように、ログリストに設定された取り込み時間の間隔でログ情報が収集されるが、この取り込み時間を変更することも可能である。変更に際し、取り込み時間をそのまま入力することも可能であるが、次のようなスケジューリング処理によって取り込み時間を決めることもできる。

【 0 0 2 6 】

図 2 1 は、スケジューリング処理のフローを示したものである。処理 2 1 1 1 でログリストからログを出力しているアプリケーションプログラム情報を取得する。処理 2 1 1 2 で、取得したアプリケーションプログラム情報をもとにアプリケーションプログラム実行環境を調べ、アプリケーションプログラムの開始時間、終了時間、実行間隔などの時間に関する制御情報を得る。これは例えば、アプリケーションプログラムが実行される装置の環境変数をenvコマンドなどで取得し、start time、 stop time、 intervalの時間設定を読み込むなどして取得することが可能である。処理 2 1 1 3 で得られた情報（開始時間、終了時間、実行間隔など）を選択肢としてユーザに提示し、ユーザの指定を受け付ける。

【 0 0 2 7 】

ユーザへの提示の例を図 2 2 に示す。この図の例では、ユーザが時間を指定してログ取得する場合、ログリストの収集時間を編集して指定する場合、それにアプリケーションの制御情報と連動させて取得する場合が示されており、この図ではアプリケーションの終了時間にログを取得することが選択されている。処理 2 1 1 3 でアプリケーションと連動させることが選択され、かつ、取得時間が指定されなかった場合はアプリケーションの終了時間を選択する。終了時間が予め明確にされていない場合には、プログラム終了時に発せられるコード（リターン値など、あるいはファイルなどの出力結果が更新された時間をもって終了時間とみなす。処理 2 1 1 4 で選択された時間をログ情報収集時間として、ログリストを書き換えて終了する。尚、ジョブプログラムについても同様である。

【 0 0 2 8 】

図 6 は、ログデータ抽出処理 2 0 7 のフローを示したものである。

【 0 0 2 9 】

この処理は、運用ルールを作成するために収集したログ情報から必要なログデータを抽出するための処理である。

【 0 0 3 0 】

処理 6 1 1 で対象となるプログラムと、このプログラムから出力されるログの書式の関連を、ログ書式ルール（図 7 参照）から読み込む。このログ書式ルール

に予めログ情報と書式の関係が記述されているかどうかを処理 6 1 2 で判定し、記述されていなかった場合には、ログ情報を取り込んだサーバから、指定されたプログラムが使用するログ書式設定情報を取り込む（6 1 3）。6 1 3 で取得されたプログラムと出力されるログ書式の関係をログ書式ルールに追加し（6 1 4）、ログ出力書式設定情報に従ってログデータをログ情報から抽出して内容を読み込み、メモリまたはバッファに格納する（6 1 5）。また処理 6 1 2 で予め関係が記述されていた場合には処理 6 1 5 に移ってデータの読み込みを行う。

【0 0 3 1】

図 7 にログ書式ルールの一例を示す。ここでは例えばプログラム 1 が出力するログの書式は書式 1 であって、書式 1 とは、最初の 0 カラム～2 5 6 カラムはコメント、次の 3 2 バイトがプログラム開始時間、次の 3 2 バイトがプログラム実行ユーザ識別子、次の 3 2 バイトが実行ジョブプロセス識別子、改行コードが入って、次の 1 6 バイトがコマンド実行開始時間、等の情報が続くことを示している。

【0 0 3 2】

図 8 は加工処理 2 0 8 のフローを示したものである。

【0 0 3 3】

処理 2 0 3 で抽出されたログデータの内容を読み込み、これより多次元テーブル（図 9 参照）を作成する（8 1 1）。

【0 0 3 4】

図 9 は、作成された多次元テーブルの一例を示したものである。この例ではデータ項目は、開始時刻、ユーザ識別子、サービス識別子、オブジェクト識別子、コマンド、ステータスによって多次元テーブルが作成されている。

【0 0 3 5】

処理 8 1 2 で作成された多次元テーブルに欠落しているデータ項目があるかどうかを判定し、欠落していた場合、処理 8 1 3 で補完が可能かどうかを判定し、可能であれば処理 8 1 4 でデータ補完処理を行う。例えば、ある行のサービス識別子が欠落していた場合、他の多次元テーブルを参照し、同じコマンド、同じオブジェクト識別子を検索する。同じコマンド、同じオブジェクト識別子がある場

合には、そのコマンド、オブジェクト識別子のサービス識別子を補完する。また、同じサービスを行った他の多次元テーブルを参照することで、一連の処理に欠落がないかを確認することができ、補完することができる。

【0036】

このように作成された多次元テーブルをメモリ、またはバッファに出力して格納する（815）。データ項目が欠落していない場合、また補完が不可能な場合はそのまま処理815へ移る。

【0037】

図10は、分析処理209のフローを示したものである。

【0038】

処理1011で多次元テーブルをメモリ、またはバッファから読み込む。処理1012で提示条件の指定を受け付ける。

【0039】

なお、指定を受け付ける代わりに、この提示条件が予め設定されていてもよい。提示指定条件が変更されると（1013）、それにもなって指定されたキーまたはサービスで多次元テーブルを並び替えて、ビュー表示テーブル（図11参照）を得る（1014）。得られたビュー表示テーブルを基に分析結果のビューの提示を行う（1015）。この分析結果の表示は、対応するアプリケーションを実行することにより行う。なお処理Aは後で説明する最適化ルールの作成処理である。以下終了が指示されるまでビューの提示条件指定を受け付け、処理1013～1016を繰り返す。

【0040】

図11にビュー表示テーブルの一例を示す。このビュー表示テーブルは、指定されたキーがユーザ識別子だった場合を示している。

【0041】

図12にビュー表示テーブル他の例を示す。このビュー表示テーブルは、サービスによる並び替えが指定された場合である。尚、ここでサービスとは一つのアプリケーションによる処理、複数のアプリケーションによる処理のいずれも含むものである。

【 0 0 4 2 】

図 1 3 ～ 図 1 6 は、処理 1 0 1 5 で表示されるビューの例を示したものである。

【 0 0 4 3 】

図 1 3 (a) は、ユーザの操作履歴を表示した例を示したものである。これは、指定条件をユーザ、履歴とした場合の表示例である。指定条件であるユーザによって、ビュー表示テーブルが生成される。また、指定条件である履歴によって、ビュー表示テーブルの時間とコマンドとをグラフによって表示するアプリケーションプログラムが選択され表示されたものである。

【 0 0 4 4 】

この操作履歴によると、ユーザがサービス 1 を開始し、文書を開く、文書を複製するなどの一連の操作を行った後、エラーが発生して、しばらく操作を停止し、その後、サービス 2 を開始していることが表示されている。

【 0 0 4 5 】

図 1 4 (a) は、ユーザの操作コマンドの累積成功率 (%) を示したものである。

【 0 0 4 6 】

これは、指定条件をコマンド、サービス、累積とした場合の表示例である。指定条件であるコマンドを第 1 優先、サービスを第 2 優先として、ビュー表示テーブルが生成される。また、指定条件である累積によって、ビュー表示テーブルのコマンドとサービスからグラフを表示するアプリケーションプログラムが選択され表示されたものである。

【 0 0 4 7 】

このグラフによると、ユーザはコマンド 3 に関してはサービス 1 において成功率が高いが、コマンド 4 に関してはサービス 1 よりもサービス 2 の成功率が高いことが示されている。

【 0 0 4 8 】

図 1 5 (a) は、ある時間帯のサービスの混み具合を示している。これは、指定条件を時間帯、サービス、累積とした場合の表示例である。指定条件である時

間を第1優先、サービスを第2優先として、ビュー表示テーブルが生成される。また、指定条件である累積によって、ビュー表示テーブルの指定された時間帯のユーザのアクセス数とサービスとの関係を表示するアプリケーションプログラムが選択され表示されたものである。

【0049】

ここでは、込み具合が人の数によって表示されており、サービス1を提供している日本支店の端末1が、他の端末に比べて混み合っていることが示されている。

図16(a)は、異常終了メッセージの内容内訳集計(%)が示されている。これは、指定条件をエラー、端末、累積とした場合の表示例である。指定条件であるエラー(ビュー表示テーブルではステータス)を第1優先、端末を第2優先として、ビュー表示テーブルが生成される。また、指定条件である累積によって、ビュー表示テーブルの指定されたステータスと端末との関係を表示するアプリケーションプログラムが選択され表示されたものである。

【0050】

この例によると、日本支店の端末1が通信不良によって異常終了する割合が高いことが示されている。

【0051】

このようなビュー表示を行うためには、既に説明したように表示するためのアプリケーションプログラムを用意しておき、ユーザからの提示条件とアプリケーションプログラムとを対応付けておく。これにより、様々なビューを表示することができる。

【0052】

図17は、実行ルール生成処理210のフローを示したものである。

【0053】

まず既存の最適化ルール(図18参照)があればそれをメモリに読み込む(1711)。次に、分析処理(209)(図中「A」で示された部分)において、指定条件のもとで、最適化ルールに従って改善できる状況があれば、改善のためのルール追加を提示する(1712)。提示した追加ルールが承認されたかどうか

かを判定し（１７１３）、承認されていれば最適化ルールに追加して（１７１４）、分析処理に戻る。承認されなければ追加せずに分析処理に戻る。

【 0 0 5 4 】

次に、最適化ルールの作成について図１３～図１６および図２４～図２７を使って説明する。

【 0 0 5 5 】

図１３（ａ）ではユーザがサービス１でエラー発生した後、サービス２に乗り換えて操作を続行している。ここでサービス１はエラー発生したままで継続されている。図１３（ｂ）は、サービス１の処理を停止するためのルールを設定するための画面の表示例を示したものである。サービス１を停止するためのルールを追加するかどうかを問合せる内容を表示し（１３０１）、「ＹＥＳ」が選択された場合には、このルールを最適化ルールに追加する場合の優先度について問合せる内容を表示する（１３０２）。この表示によって追加されたルールを図１８のルール２に示す。ルール２では、「エラー終了してから一定時間経過したサービスは自動的に停止する」と定義され、またこのルールは最優先して処理されることが定義されている。

【 0 0 5 6 】

図２４は、図１８のルール２を作成するための仕組みを示したものである。既存の最適化ルール（２４０１）は、処理１７１１で読み込む既存の最適化ルールのもっとも初期の状態をあらわしている。これは予めシステムによって定められ、あるいはユーザによって定められているものである。また一般的に汎用性のあるルールをシステムが提供し、後にユーザがカスタマイズを行ってもよい。このように予め定められた最適化ルールと、分析処理２０９で提示される条件（２４０２）とを比較し、追加ルールの案２４０３が作られる。例えば、図１３（ａ）の場合には、サービス１でエラーが発生したことがログ情報を分析した結果から得られるので、図２４の既存の最適化ルールのルール１が選択される。

【 0 0 5 7 】

これをユーザに提示したものが図１３の１３０１と１３０２になる。追加ルールがユーザによって承認されると、最適化ルール１８０１に追加され、最適化ル

ール 1 8 0 1 が更新される。このようにして、最適化ルール 1 8 0 1 は次第に詳細なルール設定を含むようになる。なお、最適化ルール 1 8 0 1 に対してはユーザが直接入力・編集作業を行っても良い。

【 0 0 5 8 】

同様にして、図 1 4 (b) の例では、コマンド 3 を成功率の高いサービス 1 で実行するよう優先し (1 4 0 1) 、コマンド 4 を成功率の高いサービス 2 で実行するよう優先する (1 4 0 2) ことをルールに追加するかどうか確認し、ルールの優先度について指定を受けている (1 4 0 3) 。 1 4 0 1 ~ 1 4 0 3 によって追加されたルールを最適化ルール (1 8 0 1) のルール 3 に示す。この時、処理 1 7 1 2 で改善のためのルールを作成提示する仕組みを図 2 5 に示す。

【 0 0 5 9 】

図 1 5 (b) の例では、1 5 0 1 で端末 1 の処理を、サービス 1 を提供する他の端末に分配するかどうかを確認し、1 5 0 2 でこのルールの優先度について確認している。1 5 0 1 と 1 5 0 2 の結果追加されたルールを最適化ルール (1 8 0 1) のルール 4 に示す。この時、処理 1 7 1 2 で改善のためのルールを作成提示する仕組みを図 2 6 に示す。

【 0 0 6 0 】

図 1 6 (b) の例では、1 6 0 1 で通信不良の多い端末 1 のジョブ実行先を他の実行先に振り替えるかどうか確認し、1 6 0 2 でこのルールの優先度について指定を受けている。1 6 0 1 と 1 6 0 2 によって追加されたルールを最適化ルール (1 8 0 1) のルール 5 に示す。この時、処理 1 7 1 2 で改善のためのルールを作成提示する仕組みを図 2 7 に示す。

【 0 0 6 1 】

以上のようにして作成された最適化ルールを基にして、サーバの運用を制御する最適化指示を生成し、発行する。

【 0 0 6 2 】

図 1 9 は、サーバの運用を制御する最適化指示の生成、発行を行う最適実行処理 2 1 1 のフローを示したものである。

【 0 0 6 3 】

まず処理 1911 で最適化ルールをメモリに読み込み、各最適化ルールからジョブ実行サーバ、アプリケーションサーバが実行できる最適化指示 a（制御コマンド）を作成する（1912）。例えば、アプリケーションを停止するコマンドや、ジョブを優先して実行するコマンドなどである。このコマンドは、最適化ルールと対応付けられており、選択された最適化ルールからコマンドを読み出すことで最適化指示 a が作成される。尚、最適化ルール及びコマンドは自由に追加することができる。

【0064】

次に、作成した最適化指示 a に矛盾が出ないか確認する（1913）。この矛盾とは、例えばジョブ実行サーバに対してジョブ A を最優先して実行するといった最適化指示が既に送られている場合に、ジョブ B を最優先して実行するといった最適化指示が作成された場合である。また、別の例としてはジョブ A を停止するといった最適化指示とジョブ A を開始するといった最適化指示が作成された場合である。これを避けるために、図 28（a）に示すようなテーブルを作成する。これは、同時に存在してはならない最適化指示を対応付けたものである。ここで、「#」となっているのは、作成された最適化指示の内容と一致することを意味している。例えば、作成された最適化指示 a が「ジョブ A を停止」となった場合、サーバ、システムの欄の「ジョブ # 開始」の「#」は「A」となることを意味する。従って、このテーブルから「ジョブ A 停止」と「ジョブ A 開始」が同時に存在してはならないことになる。

【0065】

テーブルの「サーバ」の欄は、作成された最適化指示 a が送られるサーバに対応するもので、「システム」の欄は分散システム管理サーバが接続可能なサーバ全体に対応するものである。例えば、「ジョブ A を停止」とする最適化指示 a が作成された場合、この最適化指示 a と「ジョブ A を開始」とする最適化指示 b とが同一のサーバで存在してはならないことを示す。同じく、分散システム管理サーバが接続可能なサーバ全体で「ジョブ A を開始」とする最適化指示 b と、「ジョブ A を停止」とする最適化指示 a とが存在してはならないことを示している。これは、システム中のいずれか一つのサーバに「ジョブ A を開始」とする最適化

指示 b があれば、「ジョブ A を停止」といった最適化指示 a を発行することができないことを示している。

【 0 0 6 6 】

一方、分散システム管理サーバでは、それぞれのサーバへ送った最適化指示を図 2 8 (b) のように記憶している。例えば、図 2 8 (b) ではサーバへ、ジョブ A を開始する最適化指示 b が発行されている。従って、図 2 8 (a) からジョブ A を停止する最適化指示 a はジョブ実行サーバ 1 0 4 へ送れない。従って、処理 1 9 1 2 により、ジョブ実行サーバ 1 0 4 に対し、ジョブ A を停止する最適化指示 a が生成された場合、処理 1 9 1 3 で矛盾が生じたと判断される。

【 0 0 6 7 】

矛盾がある場合には、ルールの優先度重みに従い指示を整理する (1 9 1 4) 。つまり、先の例ではジョブ A を開始するのか、ジョブ A を停止するのか優先度重みによって決定する。優先度重みは、既に図 1 3 ～図 1 6 に示すように入力されているので、これにより決定する。決定した優先度に従って、最適化指示 c を対象となるジョブ実行サーバ、アプリケーションサーバに対して与える指示を作成する (1 9 1 5) 。先の例では、「ジョブ A を停止」する最適化指示 a が作成された。この「ジョブ A を停止」する最適化指示 a が、既に発行された「ジョブ A を開始」する最適化指示 b よりも優先度が高い場合には、「ジョブ A を開始」する最適化指示 b を削除する新たな最適化指示 c を作成する。この新たな最適化指示 c をサーバに送ることで「ジョブ A を開始」する最適化指示 b が削除される。この後に、既に作成した「ジョブ A を停止」する最適化指示 a をサーバに送ることで、ジョブ A を停止することができる。

【 0 0 6 8 】

作成された最適化指示 a、最適化指示 c は、ネットワークを介してジョブ実行サーバ、アプリケーションサーバに発行される (1 9 1 6) 。最適化指示 c については、図 2 8 (a) のテーブルのサーバの欄に最適化指示 b が記載されている場合には、最適化指示 a と最適化指示 c とを同一のサーバへ送る。一方、図 2 8 (a) のテーブルのシステムの欄に最適化指示 b が記載されている場合には、最適化指示 c を分散システム管理サーバが接続可能なサーバへ送る。

【 0 0 6 9 】

尚、作成された最適化指示 a と既に送られている最適化指示 b とが矛盾を生じると判断され、作成された最適化指示 c の優先度の方が低い場合には、作成された最適化指示 c を発行しない。また、発行のタイミングは指定することができ、例えば、時差がある分散環境などで、業務に差し支えのない時間に発行することができる。また、分散システム管理サーバ（108）全体の処理をバッチ処理などで行うことが可能で、システムに負荷にならないように管理することができる。尚、例としてジョブ実行サーバについて説明したがアプリケーションサーバも同様にして最適化指示が発行される。このように、実行ルール生成処理では、運用ルールに基づいて、サーバを制御する制御コマンドを発行する。これにより、システム全体を効率よく運用させることができる。また、分散システム管理サーバに、同時に実行できないコマンドを登録することができ、実行ルール生成処理では、実行させるべきコマンドを残すように最適化指示を発行する。これにより、サーバ、あるいはシステム全体を効率よく運用することが可能となる。

【 0 0 7 0 】

以上説明してきたように、本実施例においては、サービスの運用を定めた複数の運用ルールと、それぞれの運用ルール毎に実行するための条件が設定されており、サービスの実行により生成されたログ情報をそれぞれのサーバから収集し、収集したログ情報が条件を満たした場合に、この条件に対応付けられた運用ルールに基づいてサーバを制御する命令を出力する。これにより、アプリケーションを作り変えることなく、既存のアプリケーションが予め持っている既存のログ情報を流用して分散環境におけるジョブ実行の状況を分析してユーザが把握することができる。

【 0 0 7 1 】

また、ログの収集、解析はシステムの動作とは関係なく処理を行うことができるので、システムにネットワーク負荷の影響を与えずに最適化処理を行うことができる。

【 0 0 7 2 】

また、過去から累積した履歴の分析とそれを基にしたシステム改善提案を行う

ことのできるシステム管理支援方法を提供することができる。

【 0 0 7 3 】

【発明の効果】

本発明によれば、効率よく運用できる分散システム的环境を提供することができる。

【図面の簡単な説明】

【図 1】

システム全体の構成を示した図である。

【図 2】

システム管理支援処理部の記憶部の構成を示した図である。

【図 3】

アプリケーションサーバの記憶部の構成を示した図である。

【図 4】

ジョブ実行サーバの記憶部の構成を示した図である。

【図 5】

ログ検出処理部のフローを示した図である。

【図 6】

ログデータ抽出処理部のフローを示した図である。

【図 7】

ログ書式ルールの一例を示した図である。

【図 8】

抽出データ加工処理部のフローを示した図である。

【図 9】

多次元テーブルの一例を示した図である。

【図 1 0】

分析処理のフローを示した図である。

【図 1 1】

ビュー表示テーブルの一例を示した図である。

【図 1 2】

ビュー表示テーブルの一例を示した図である。

【図 1 3】

ビュー提示の一例を示した図である。

【図 1 4】

ビュー提示の一例を示した図である。

【図 1 5】

ビュー提示の一例を示した図である。

【図 1 6】

ビュー提示の一例を示した図である。

【図 1 7】

実行ルール生成処理のフローを示した図である。

【図 1 8】

最適化ルールの一例を示した図である。

【図 1 9】

最適化実効処理のフローを示したものである。

【図 2 0】

ログ情報取り込み処理のフローを示したものである。

【図 2 1】

スケジューリング処理のフローを示したものである。

【図 2 2】

スケジューリング処理のユーザへの定時の一例を示したものである。

【図 2 3】

収集先ログ情報リストの一例を示したものである。

【図 2 4】

ビュー提示の場合のルールの一例を示したものである。

【図 2 5】

ビュー提示の場合のルールの一例を示したものである。

【図 2 6】

ビュー提示の場合のルールの一例を示したものである。

【図 2 7】

ビュー提示の場合のルールの一例を示したものである。

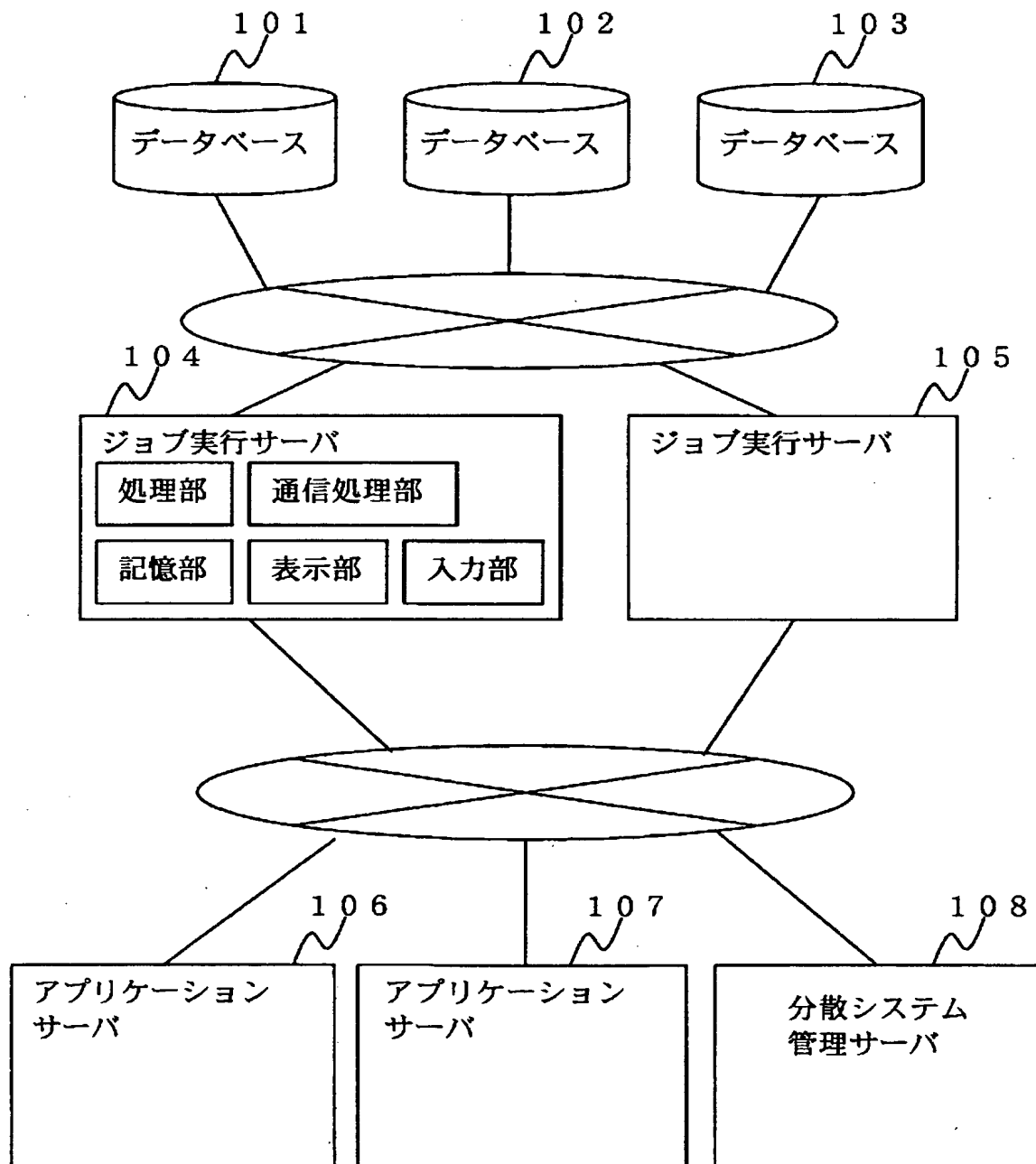
【符号の説明】

- 1 0 1 データベース
- 1 0 2 データベース
- 1 0 3 データベース
- 1 0 4 ジョブ実行サーバ
- 1 0 5 ジョブ実行サーバ
- 1 0 6 アプリケーションサーバ
- 1 0 7 アプリケーションサーバ

【書類名】 図面

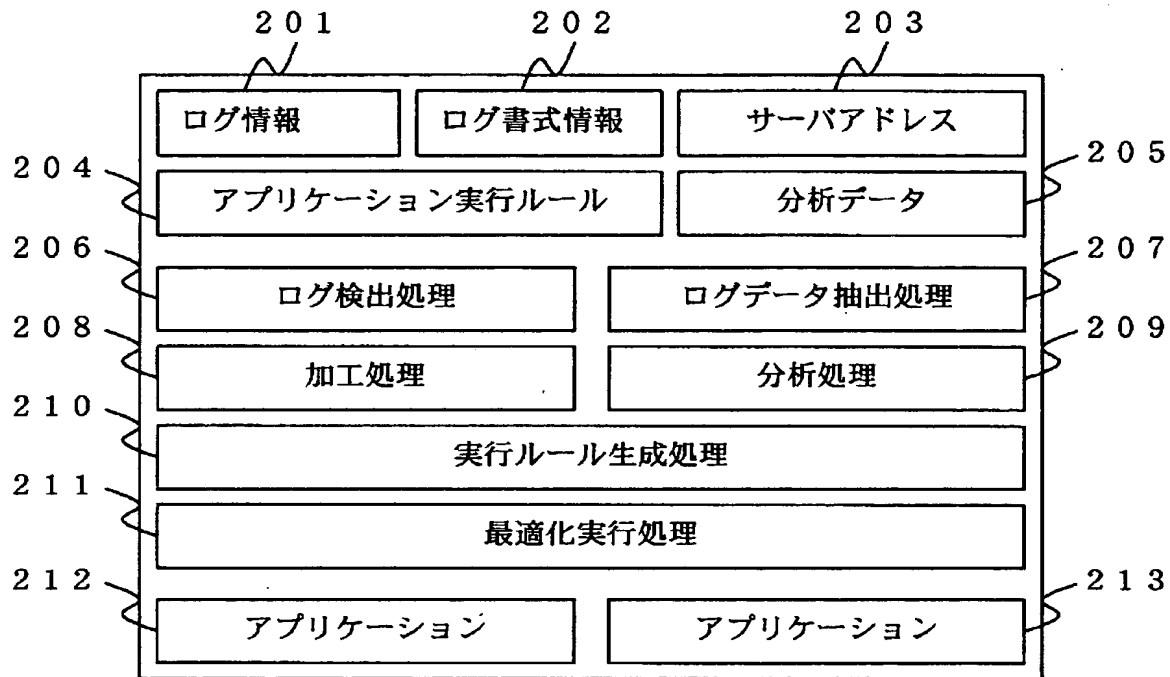
【図1】

図1



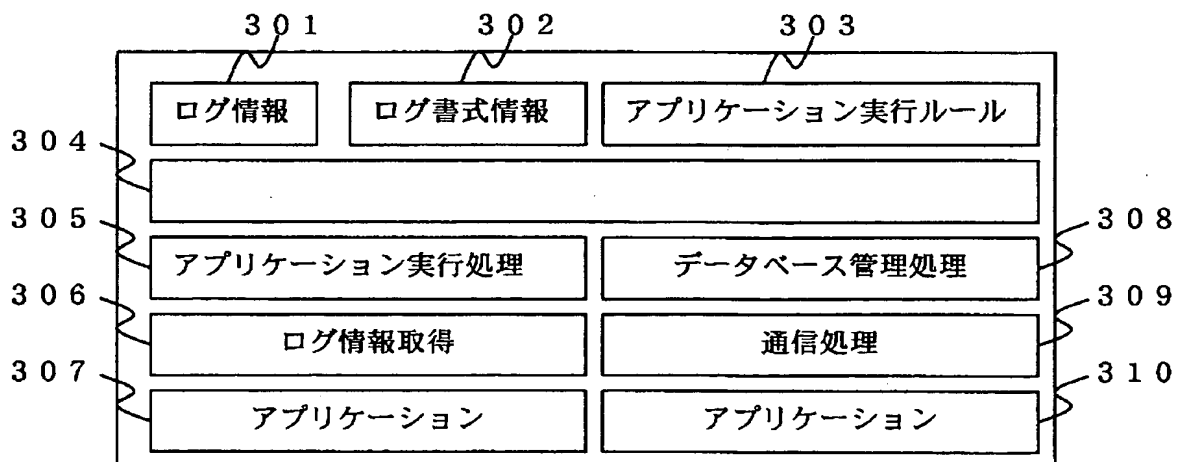
【図 2】

図 2



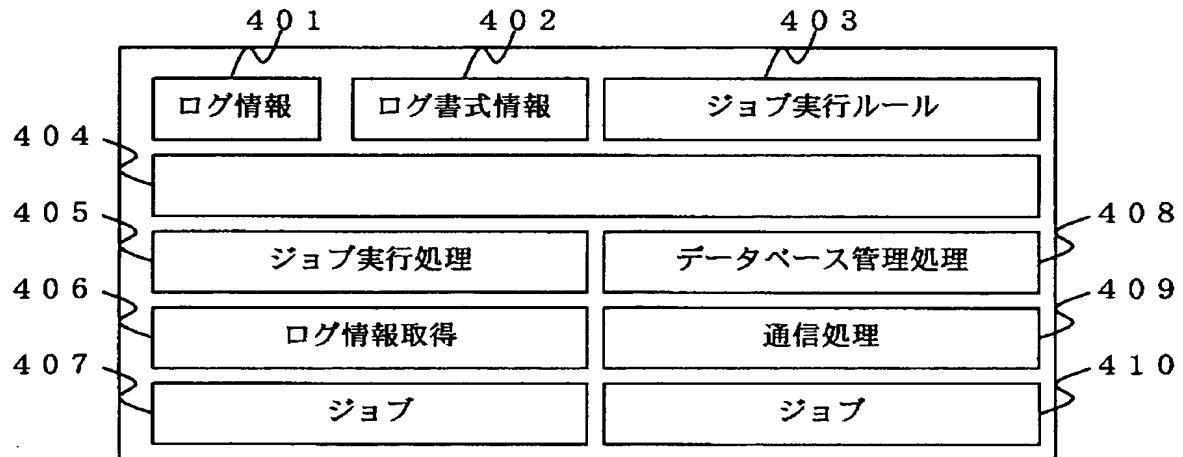
【図 3】

図 3



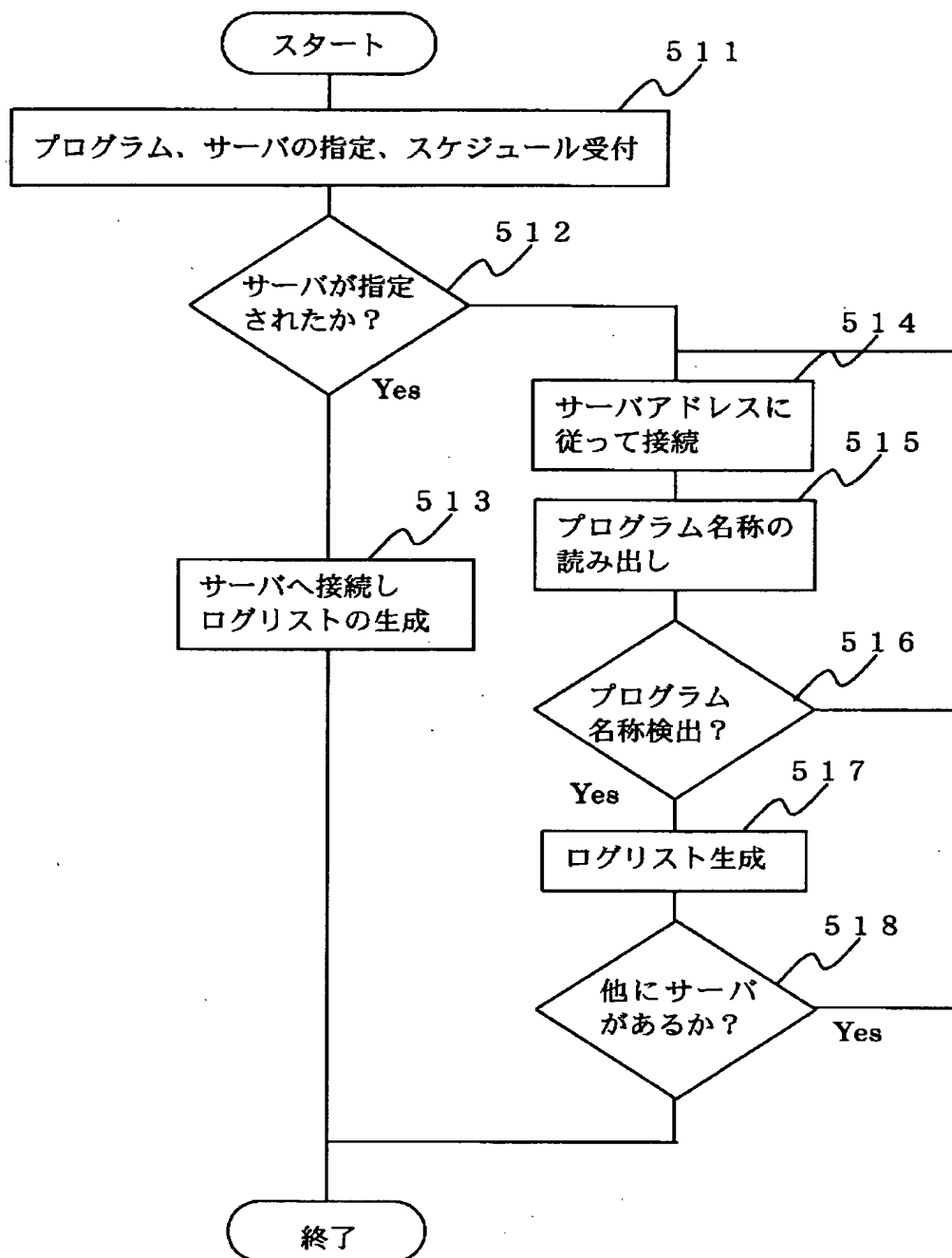
【図 4】

図 4



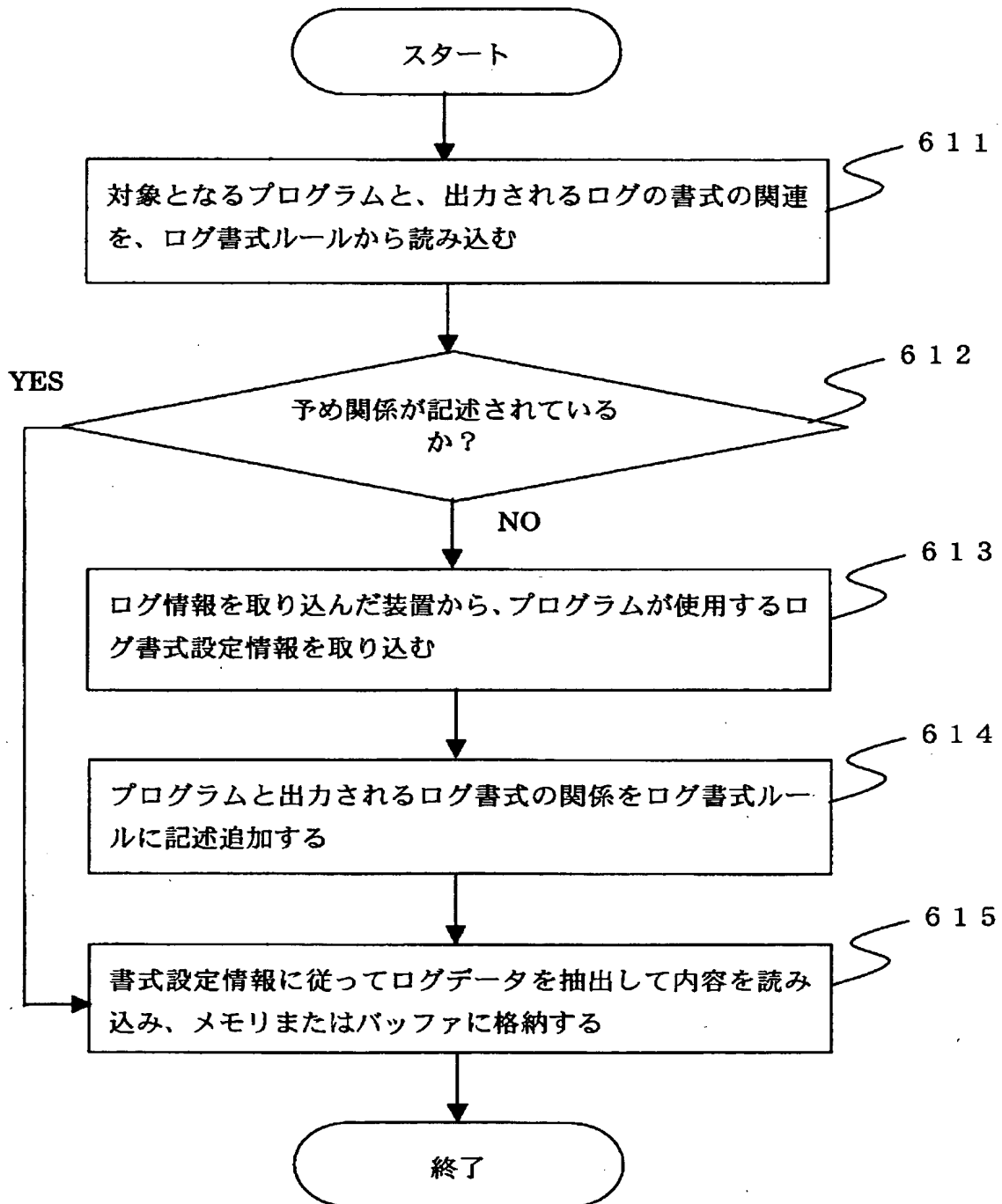
【図 5】

図 5



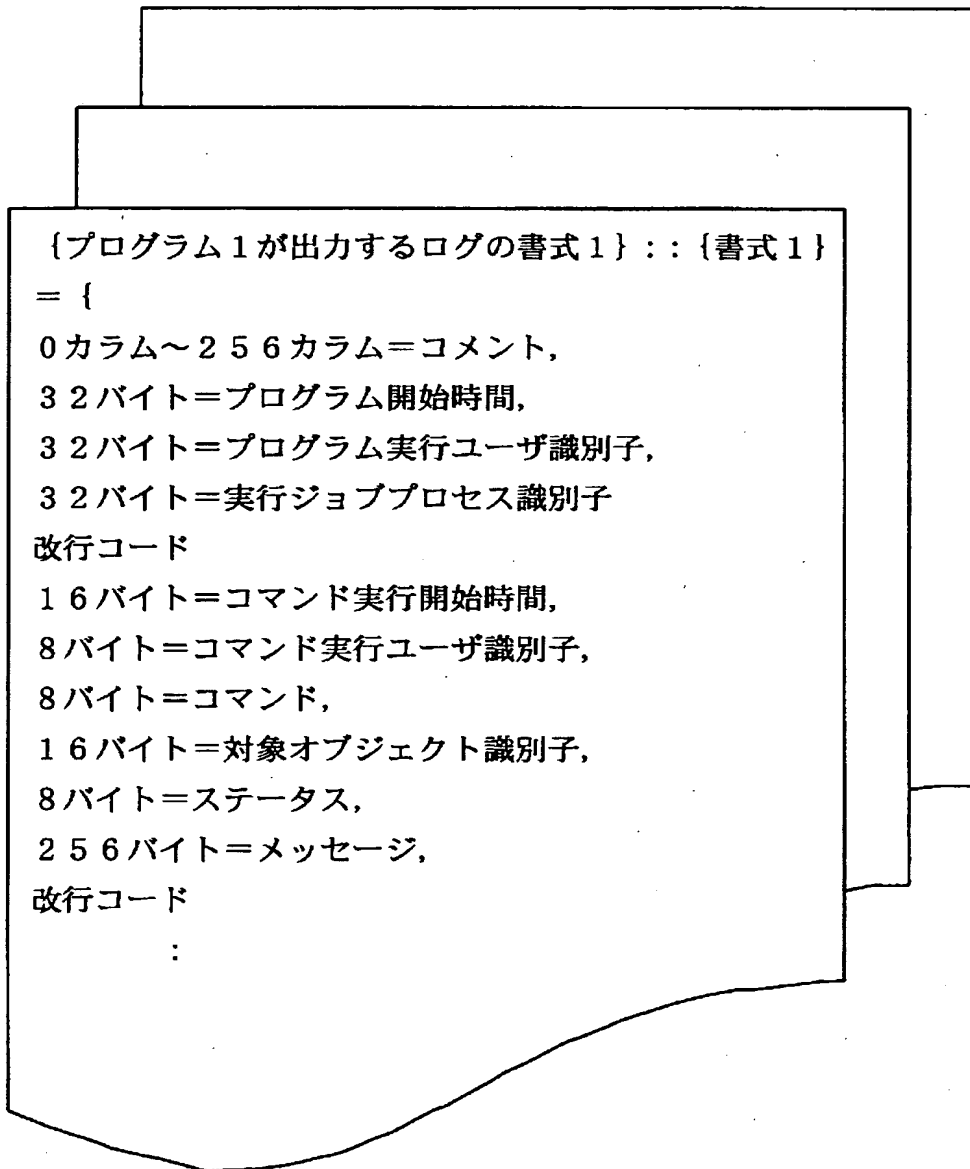
【図 6】

図 6



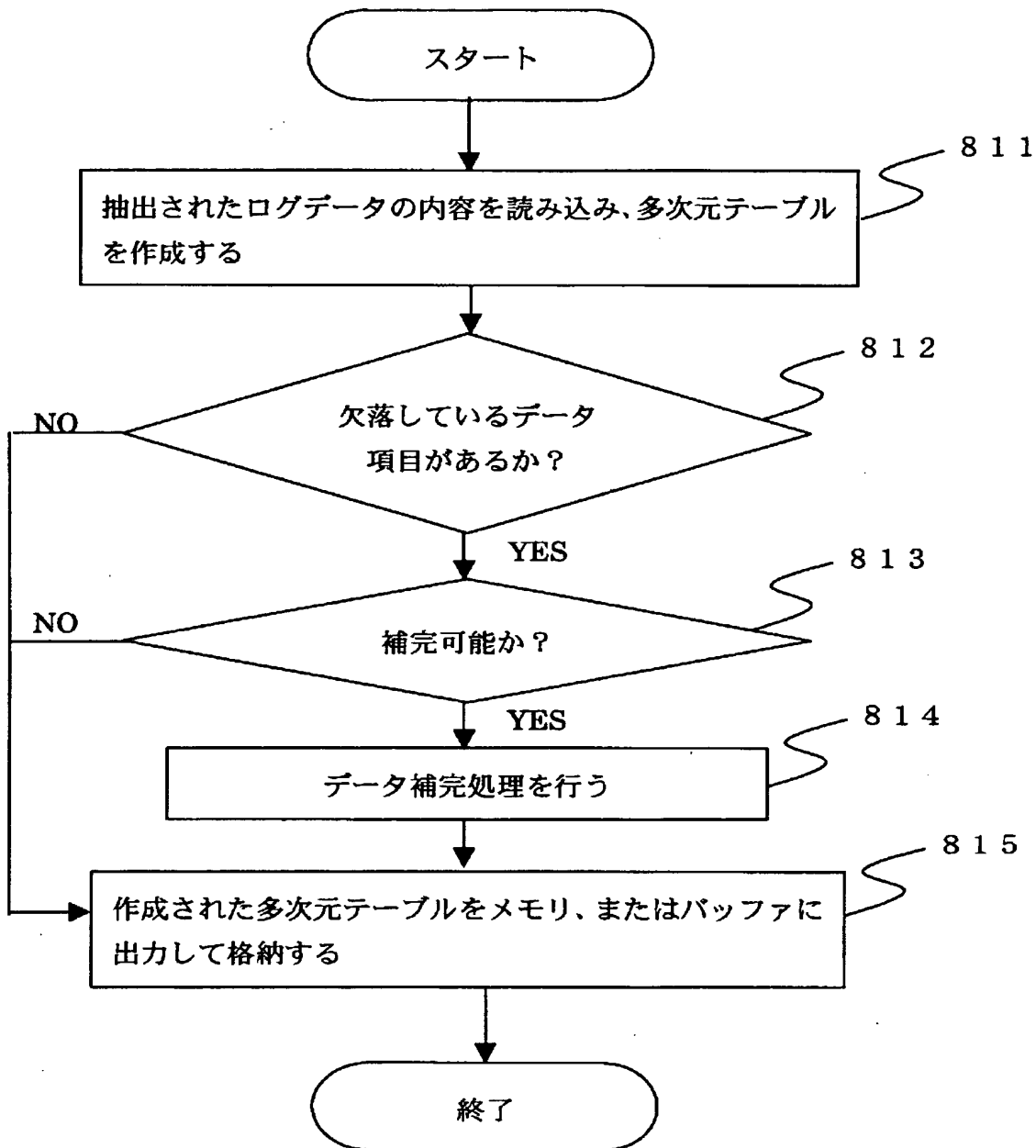
【図 7】

図 7



【図 8】

図 8

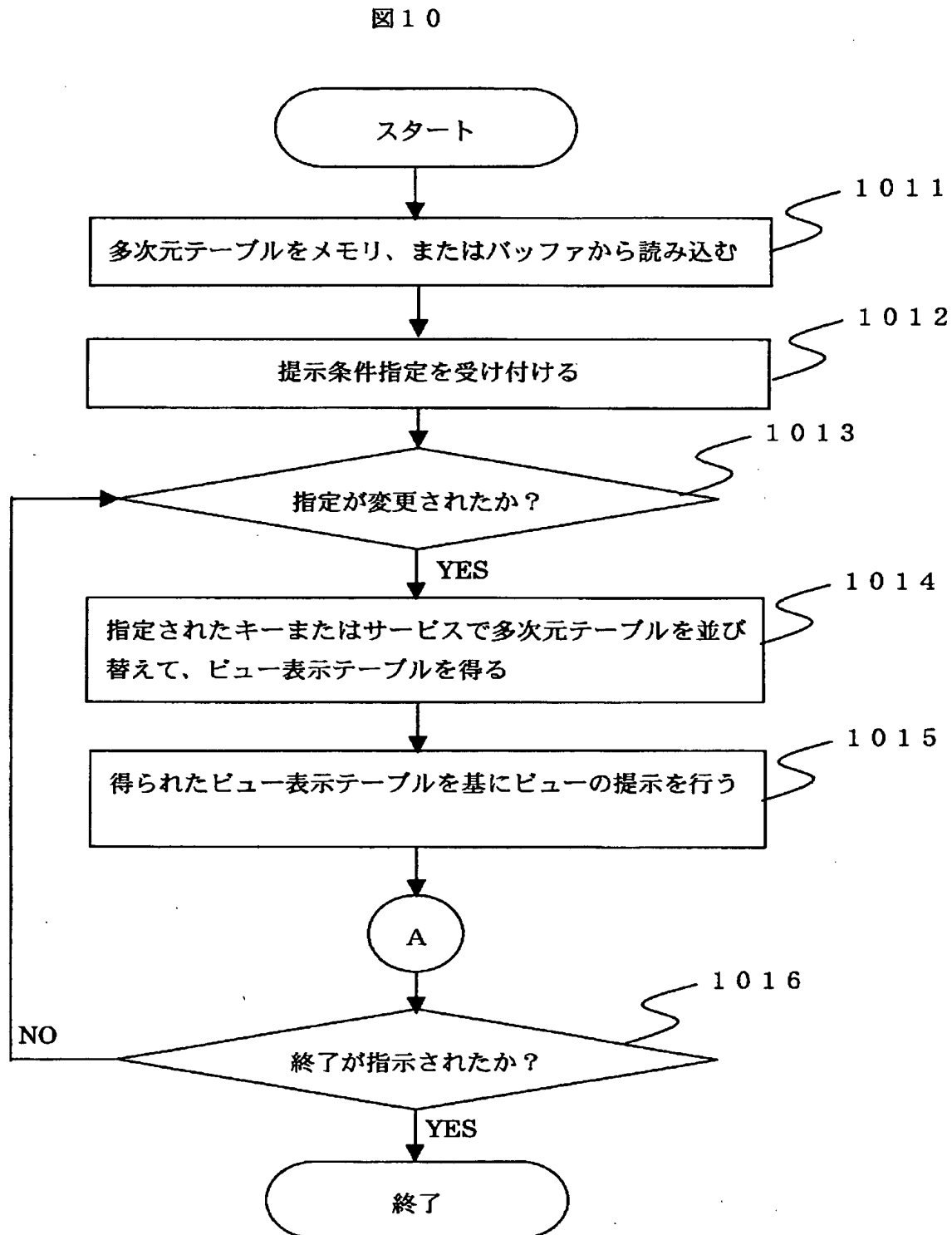


【図 9】

図 9

開始時刻	ユーザ 識別子	サービス 識別子	オブジェクト 識別子	コマンド	ステータス
00 : 00 : 00	User1	Service1	Object1	Command1	Status1
00 : 00 : 25	User1	Service1	Object1	Command2	Status2
00 : 00 : 29	User2	Service2	Object2	Command1	Status3
00 : 01 : 05	User1	Service2	Object3	Command3	Status4

【図10】



【図 1 1】

図 1 1

ユーザ 識別子	開始時刻	サービス 識別子	オブジェクト 識別子	コマンド	ステータス
User1	00:00:00	Service1	Object1	Command1	Status1
User1	00:00:25	Service1	Object1	Command2	Status2
User1	00:01:05	Service2	Object3	Command3	Status4

ユーザ 識別子	開始時刻	サービス 識別子	オブジェクト 識別子	コマンド	ステータス
User2	00:00:29	Service2	Object2	Command1	Status3

【図 1 2】

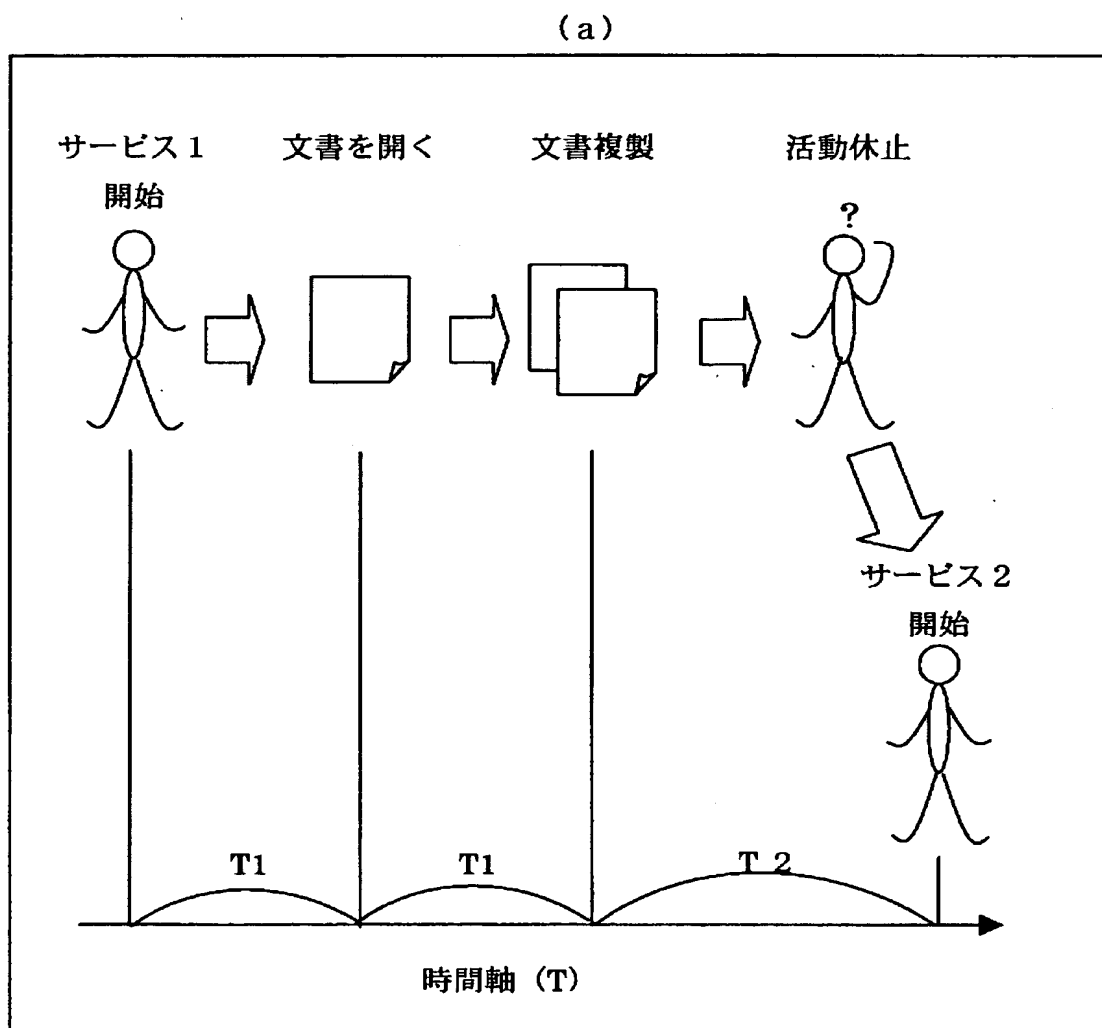
図 1 2

サービス 識別子	開始時刻	ユーザ 識別子	オブジェクト 識別子	コマンド	ステータス
Service1	00:00:00	User1	Object1	Command1	Status1
Service1	00:00:25	User1	Object1	Command2	Status2

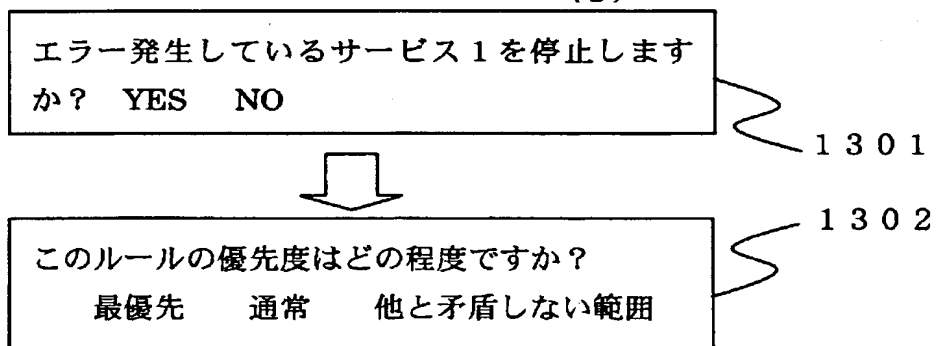
サービス 識別子	開始時刻	ユーザ 識別子	オブジェクト 識別子	コマンド	ステータス
Service2	00:00:29	User2	Object2	Command1	Status3
Service2	00:01:05	User1	Object3	Command3	Status4

【図 13】

図 13



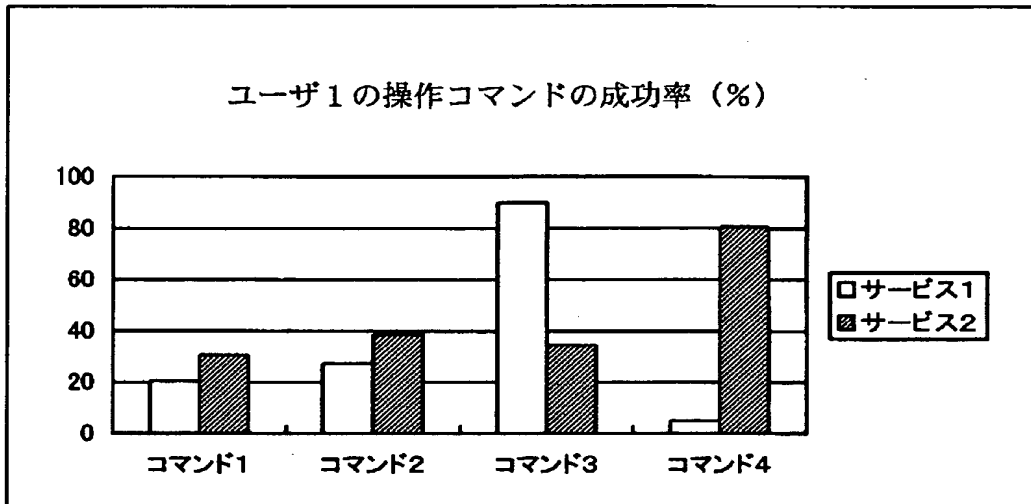
(b)



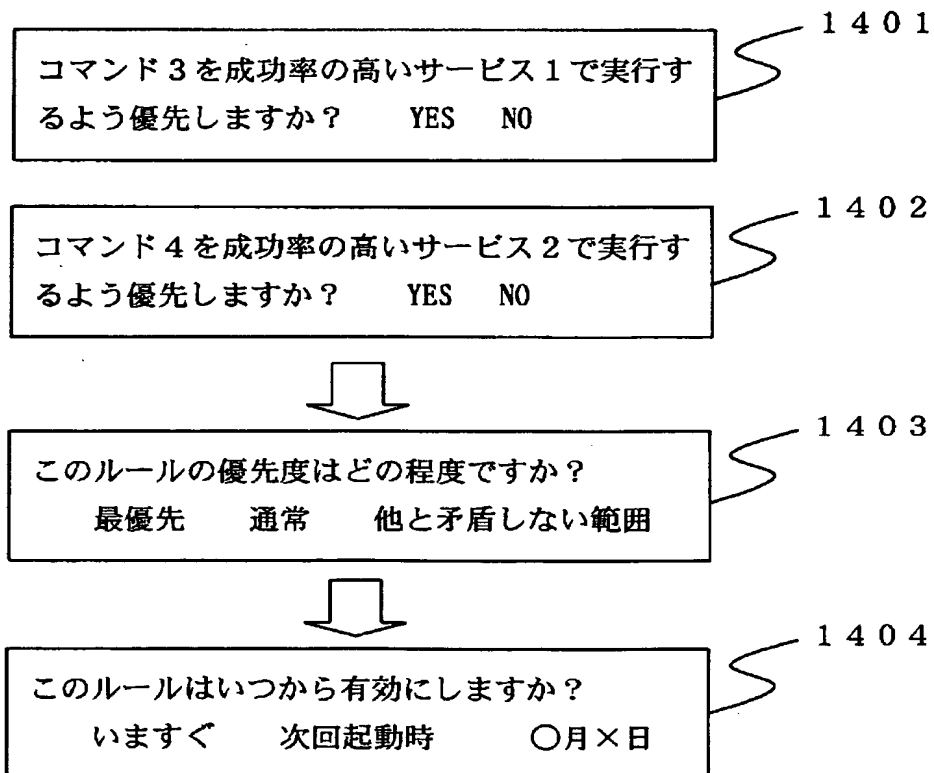
【図 1 4】

図 1 4

(a)

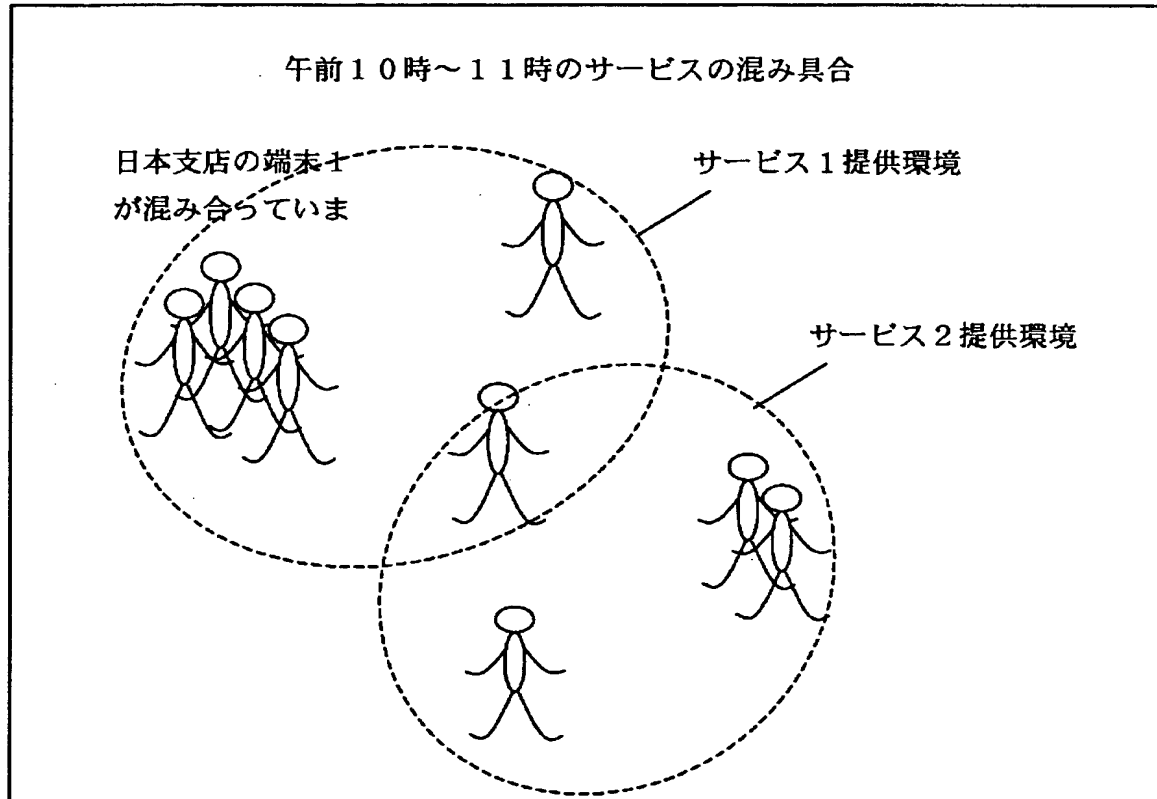


(b)

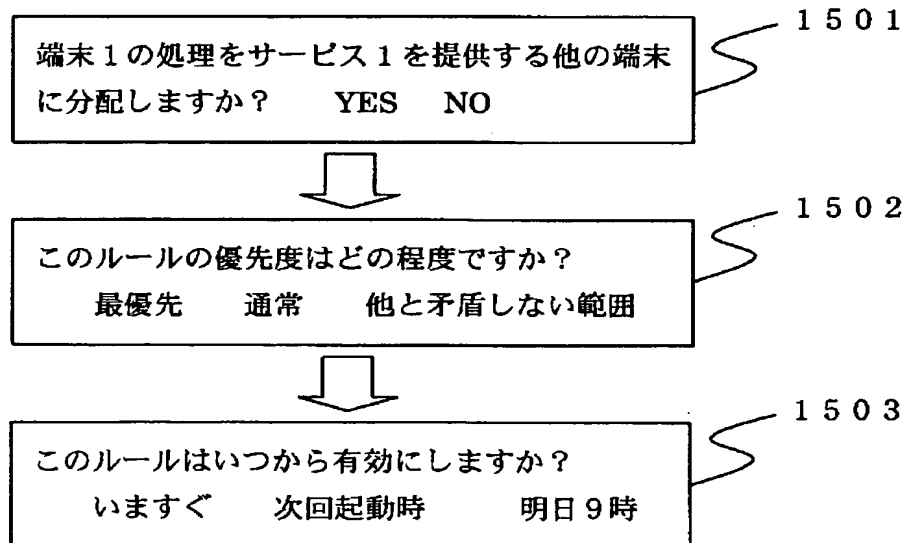


【図 15】

図 15
(a)



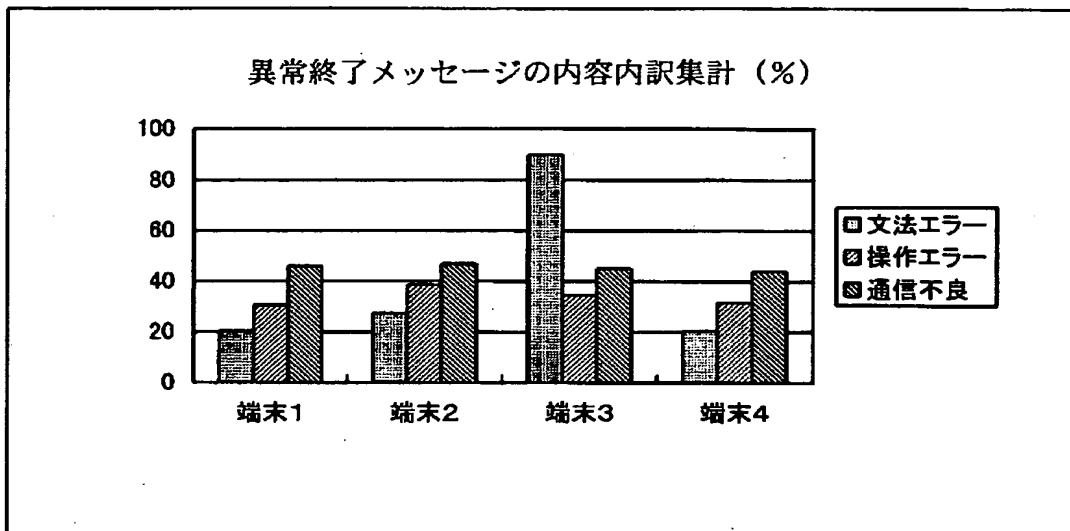
(b)



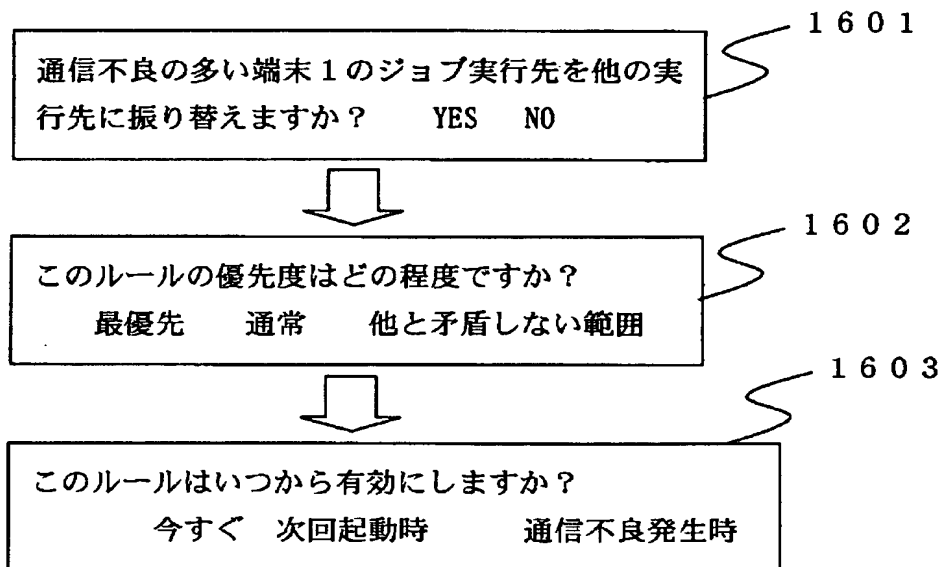
【図 1 6】

図 1 6

(a)

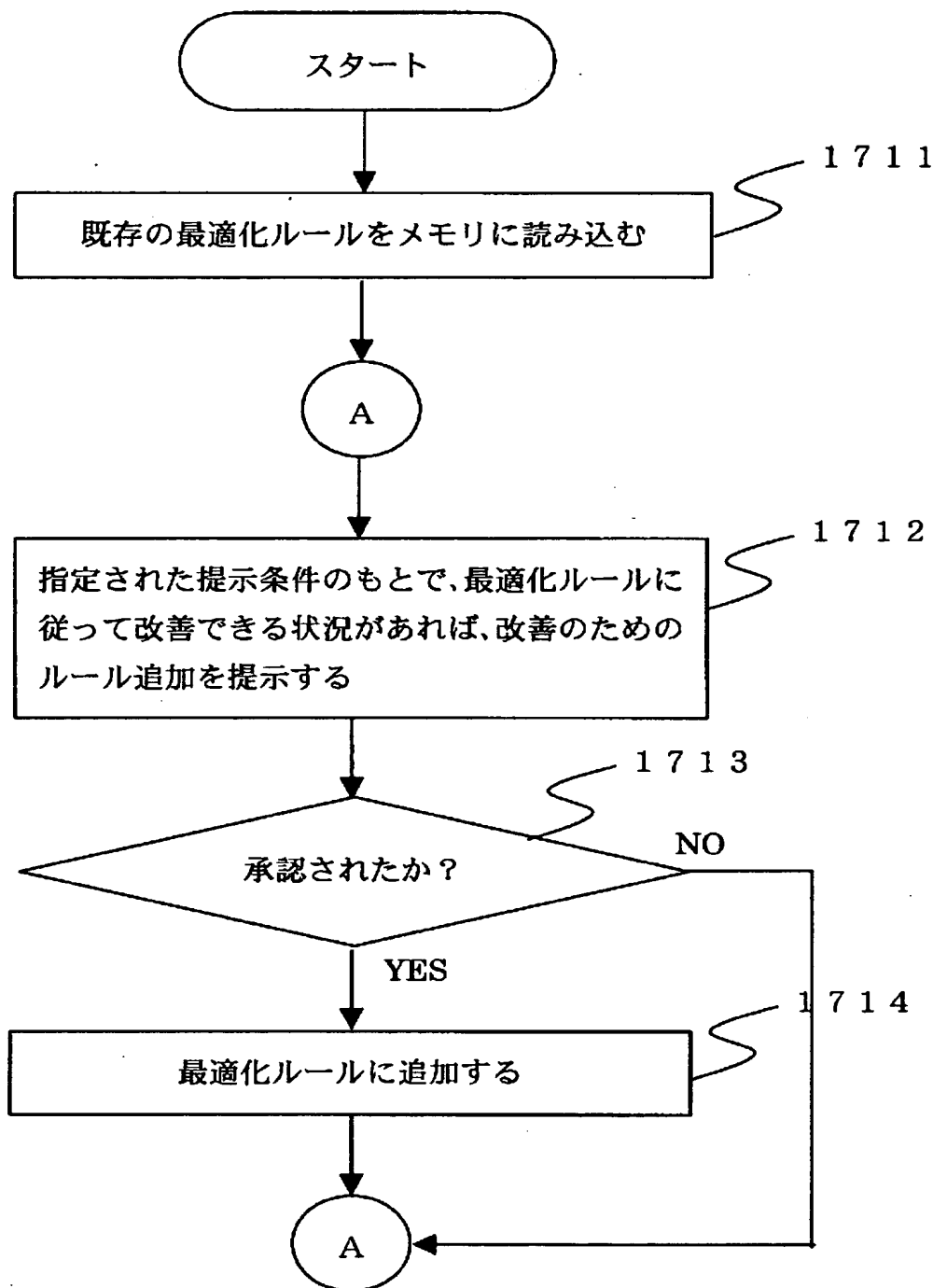


(b)



【図 1 7】

図 1 7



【図 1 8】

図 1 8

ルール 1 【他と矛盾しない範囲】：ユーザの操作時間間隔がしきい値を越えて増加したときは、操作ガイダンスを提示する

ルール 2 【最優先】：エラー終了してから一定時間経過したサービスは自動的に停止する

ルール 3 【通常】：コマンド成功率の高いサービスのコマンド発行処理を優先する

ルール 4 【通常】：同一サービスが受けられる端末のうち、ネットワーク負荷の高い端末で行われる処理を他の端末に配分する

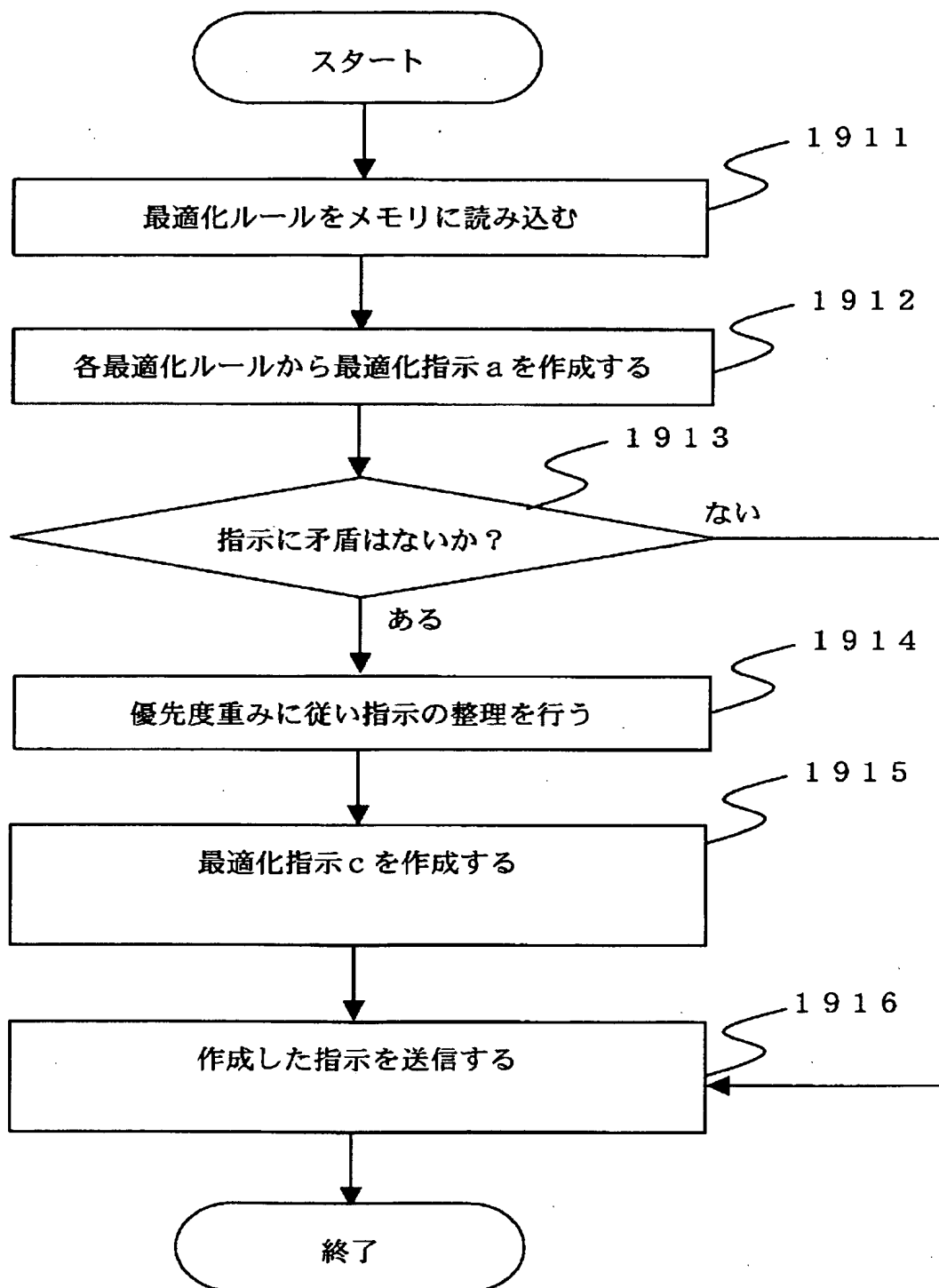
ルール 5 【最優先】：通信不良で異常終了している割合が一定値を越えた場合、ジョブ実行先を別のジョブ実行先から検索して代用する

：

：

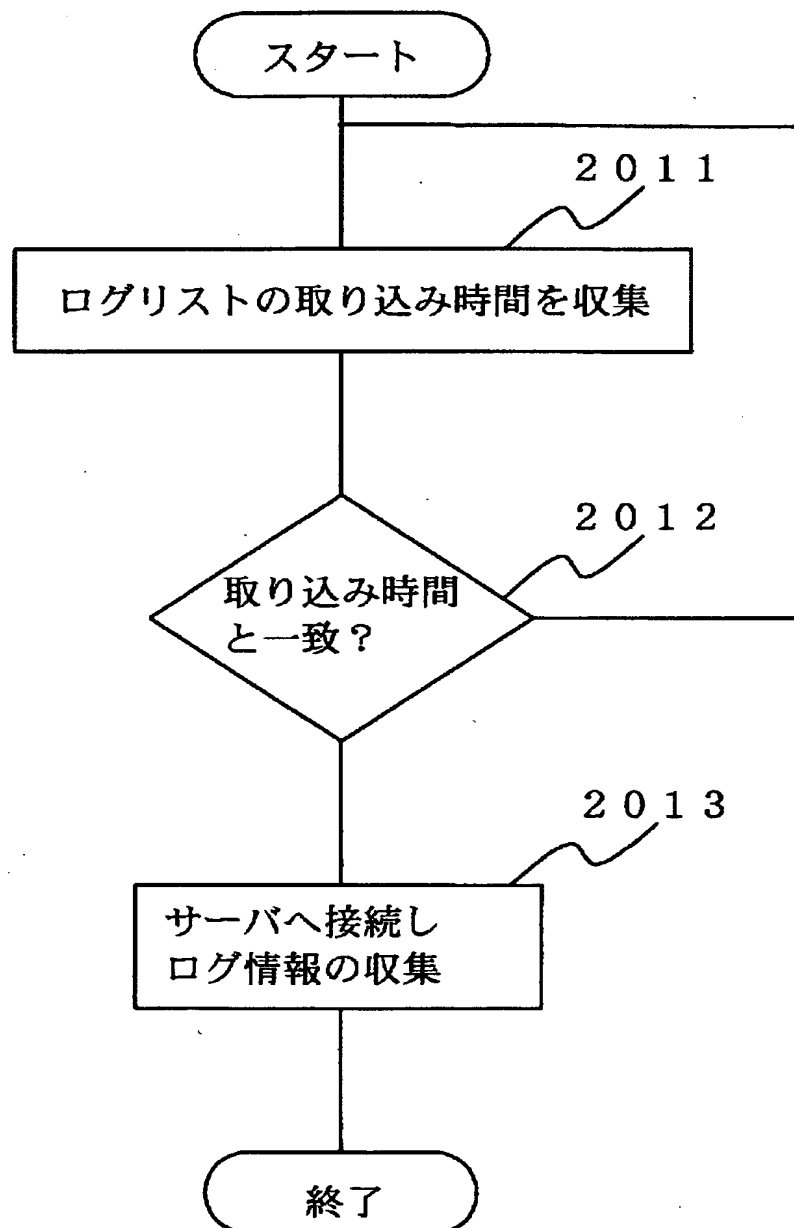
【図19】

図19



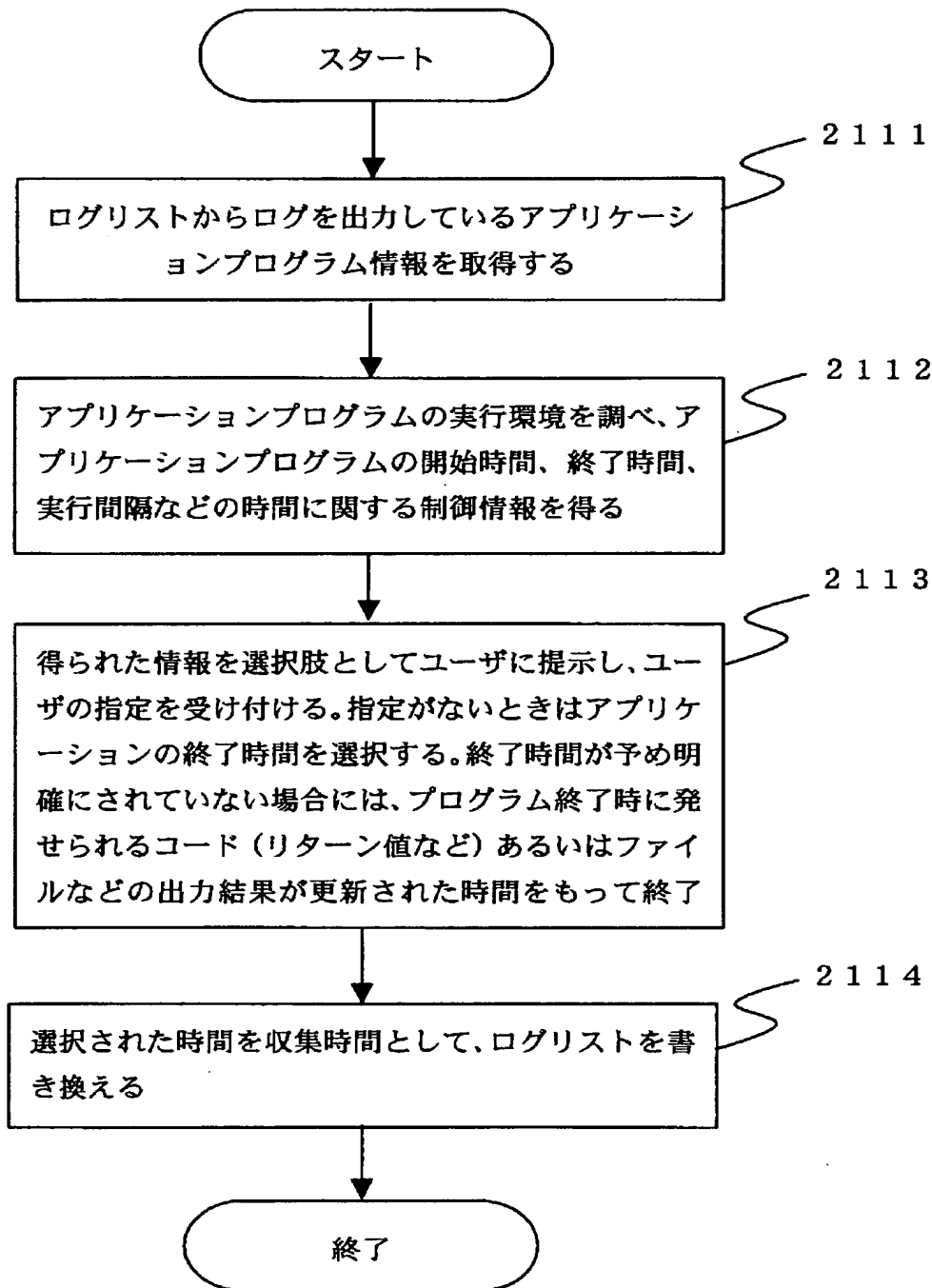
【図 2 0】

図 2 0



【図 2 1】

図 2 1



【図22】

図22

ログ収集のタイミングをスケジューリン

☐ 直接指定する

☐ ファイル編集画面へ

時間

分ごとにログ取得

☒ アプリケーションと連動させる（選択可能なものだけ表示されます）

☐ 開始時間

☒ 終了時間

☐ インターバルごと

☐ データベース更新時

【図 2 3】

図 2 3

(a) スケジューリングの指定がない場合

装置アドレスとログ情報名	アプリケーション名	取り込み時間
188.12.13.11/logfile1	188.12.13.11/App_1	デフォルト
192.11.13.32/D/Log/logfile2	202.11.34.22/App_2	デフォルト
211.00.23.323/C/log_file3	123.232.4.332/App_3	デフォルト
233.44.556.32/Temp/log_data4	233.44.556.32/App_4	デフォルト

(b) スケジューリングの指定がある場合

アドレスとファイル名	アプリケーション名	取り込み時間
188.12.13.11/logfile1	188.12.13.11/App_1	デフォルト
192.11.13.32/D/Log/logfile2	202.11.34.22/App_2	1 5 分ごと
211.00.23.323/C/log_file3	123.232.4.332/App_3	1 日ごと
233.44.556.32/Temp/log_data4	233.44.556.32/App_4	アプリケーション プログラム終了時

【図 24】

図 24

既存の最適化ルール (2401)

ルール 1 : エラー発生したとき、エラー発生を起こしたコマンド
が使用しているサービスの処理を尋ねる。
ルール 2 : 時間 T 以上経過して操作が行われなかったサービス
を停止対象とするか尋ねる。
ルール 3 : ルールには他のルールに対する優先度を指定する。

指定された提示条件 (2402)

ユーザの操作履歴マップ (提示項目=ユーザ、サービス、開始時
刻、コマンド履歴、時間経過、エラー状況) を作成して提示する。

処理 (1712) : 改善のための追加ルール作成と

エラー発生しているサービス 1 の処理は？
時間 T 以上経過して操作が行われないサービス 1 は停止する
か？

2403



ユーザへの提示例 (図 13)

エラー発生しているサービス 1 を停止します
か？ YES NO

1301

このルールの優先度はどの程度ですか？
最優先 通常 他と矛盾しない範囲

1302

【図 25】

図 25

既存の最適化ルール（2501）

ルール4：コマンド操作の成功率が20%以上異なるサービスがある場合、成功率の高いサービスを優先するか尋ねる。
ルール5：ルールの有効設定時間は、いますぐ、次回起動、日付指定、イベント指定のうちいずれかを選択する。

指定された提示条件（2502）

ユーザの操作コマンドの成功率を提示する

処理（1712）：改善のための追加ルール作成と提示

2503

コマンド3をサービス1＞サービス2で優先とするか？
優先度の設定はいつ以降有功とするか？



ユーザへの提示例（図14）

コマンド3を成功率の高いサービス1で実行するよう優先しますか？ YES NO

1401

このルールはいつから有効にしますか？

いますぐ 次回起動時 ○月×日

1404

【図 26】

図 26

既存の最適化ルール（2601）

ルール 6 : ある端末の混み具合が同一サービス内で 300% を越えるとき異常発生アラームを出し、処理を尋ねる。
ルール 7 : 異常発生時に同一サービス内で複数端末が使用可能な場合は、どの端末を使用するか尋ねる。

指定された提示条件（2602）

午前 10 時～11 時のサービスの混み具合の提示

処理（1712）：改善のための追加ルール作成と提示

2603

端末 1 に異常発生を提示し、処理を尋ねる。
同一サービス 1 内の他の端末を使用するか？



ユーザへの提示例（図 15）

端末 1 の処理をサービス 1 を提供する他の端末に分配しますか？ YES NO

1501

【図 27】

図 27

既存の最適化ルール（2701）

ルール 8：通信不良が 20%を超える端末はジョブ実行を他の端末に振り替えるか尋ねる。

ルール 5：ルールの有効設定時間は、いますぐ、次回起動、日付指定、イベント指定のうちいずれかを選択する。

指定された提示条件（2702）

異常終了メッセージの内容内訳集計（%）（提示項目：文法エラー、操作エラー、通信不良）を提示する。

処理（1712）：改善のための追加ルール作成と提示

2703

通信不良が多い端末 1 のジョブ実行を他の端末に振り替えるか？



ユーザへの提示例（図 16）

通信不良の多い端末 1 のジョブ実行先を他の実行先に振り替えますか？ YES NO

1601

1603

このルールはいつから有効にしますか？

今すぐ 次回起動時 通信不良発生時

【書類名】 要約書

【要約】

【課題】

既存のアプリケーション全体を作り直すことなく、日々変化する分散サービスに適応して、ネットワーク負荷のオーバーヘッドをかけることなく、アプリケーションの最適実行を行うことのできる分散システム管理方法を提供することである。

【解決手段】

プログラム実行時のログ情報を取得し、これを集計、加工してユーザに提示すると共に、ユーザによって指示された最適化ルールによってアプリケーション実行最適化のための改善指示の作成、発行を行う。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所

出 願 人 履 歴 情 報

識別番号 [0 0 0 2 3 3 0 5 5]

1. 変更年月日 1 9 9 0 年 8 月 7 日

[変更理由] 新規登録

住 所 神奈川県横浜市中区尾上町 6 丁目 8 1 番地

氏 名 日立ソフトウェアエンジニアリング株式会社